Evaluating 2011 Russian Duma Election for Fraud

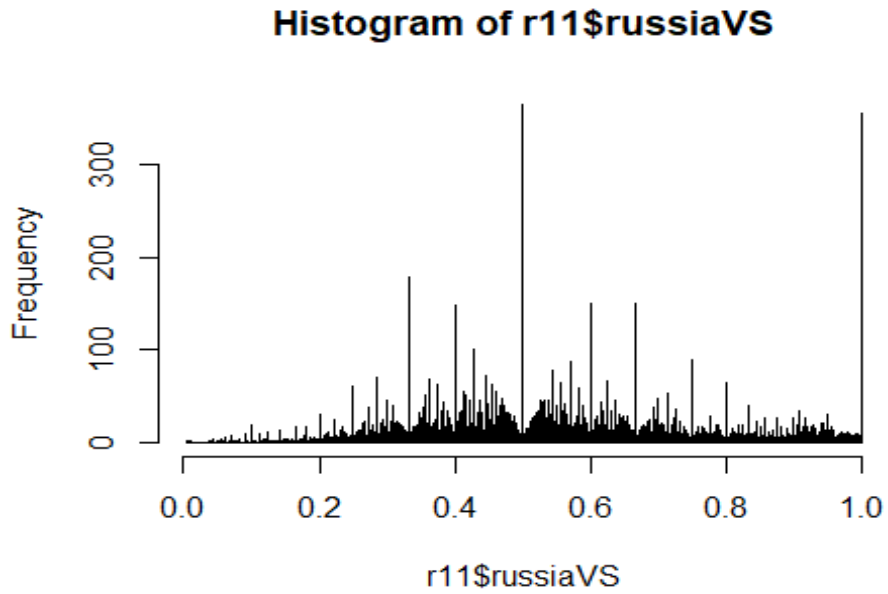
Selina Wu
April 1, 2020

```
set.seed(94304)

load ("fraud.RData")
library(MASS)
```

2011 Russia vote share

United Russia's vote share was computed as a proportion of the voters who turned out, and these proportions were plotted as a histogram, with each bar representing each of the uniquely observed fractions. The unique() function was used in order to group the observed vote share fractions into unique fractions.

```
r11 <- russia2011                    # to make it easier for me to use r11 as
data frame
r11$russiaVS <- r11$votes / r11$turn   # calculating the VS (returns decimal
in each row)

frac_RVS <- as.fractions(r11$russiaVS)
uni_frac11 <- unique(frac_RVS)   # a vector of length 68137

hist(r11$russiaVS, breaks = length(uni_frac11))
```



The histogram bars for fractions with low integers and denominators appear to peak, indicating high frequencies for these fractions.

The ten most frequently occurring fractions for the vote share are printed in their decimal forms below. The aggregate() function is used to split the resulting vote share proportions into lists based off their occurences, and the lengths of the lists for each proportion are calculated. Within the agg data frame, the order() function is used to sort the fractions by descending frequency so the highest frequency fraction is in the first row. The agg data is then subsetted to print out the ten most common fractions.

```
agg<- aggregate(r11$russiaVS, list(r11$russiaVS), length)
agg<- agg[order(agg$x, decreasing=T),]
agg[c(1:10),c(1)]

##  [1] 0.5000000 1.0000000 0.3333333 0.6000000 0.6666667 0.4000000 0.4285714
##  [8] 0.7500000 0.5714286 0.5454545
```

Probability model to investigate low fraction frequencies due to chance

A Monte Carlo simulation is conducted with 1000 simulation. A matrix is created to store the simulated vote share values for each of the precincts in each simulation. Each row represents one precinct, and each column represents one of 1000 simulations.

```
nsims <- 1000
r11_Monte <- matrix(data=NA, nrow=nrow(r11), ncol=nsims)
```

A for loop is used to run the simulation 1000 times. Within the for loop, the simulated turnout is first calculated, assuming the turnout for a precinct has a binomial distribution whose size equals the number of voters in the precinct, and whose success probability equals the turnout rate for the precinct.

The simulated vote share is also calculated within the for loop. We assume the votes for United Russia in each precinct follows a binomial distribution, conditioned on the turnout calculated right before this step. The sample size equals the number of voters who turned out in the precint, and the success probability is the observed vote share for the precinct. The number of raw votes is then divided by each precint's simulated turnout.

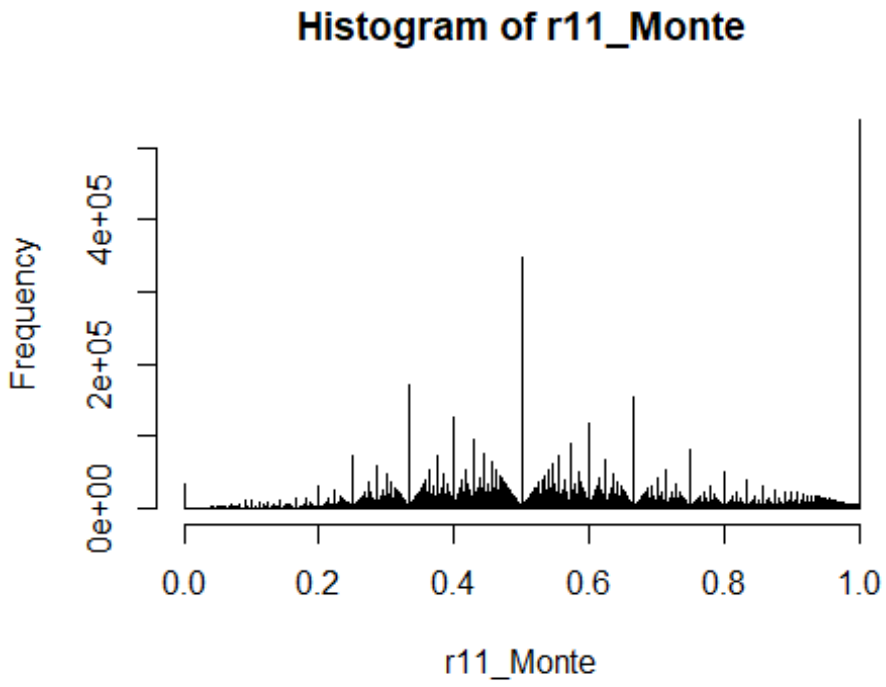
As the loop iterates, the vote share values are stored in the matrix, with each row representing the precint, and each column representing one of 1000 simulations. The code to print the matrix is commented out, as it is a matrix of dimension 94995 x 1000.

```
for(i in 1:nsims){
  #a) simulate turnout in precint using binomial distribution (returns vector
of simulated # of people to show up to vote)
  r11$prop.turnout <- (r11$turnout / r11$N)
  q2a_r11 <- rbinom(nrow(r11), r11$N, p=(r11$turnout / r11$N))

  #b) simulate observed vote share given turnout
  q2b_r11 <- rbinom(nrow(r11), q2a_r11, r11$russiaVS) / q2a_r11

  r11_Monte[,i] <- q2b_r11
```

```
}
# r11_Monte
```

A histogram of all the data from the simulations is plotted in order to show the distribution of fractions under this simulation.

```
hist(r11_Monte, breaks = length(unique(frac_RVS)))  #too many breaks when use unique fractions from the Monte Carlo simulation
```



Comparing simulation results with actual election results

COMPARING THE FREQUENCY OF 1/2

The ifelse() function is used to identify which simulated elections yielded a vote share fraction of 1/2.

```
binary_12 <- ifelse ((r11_Monte==0.50), 1, 0)
```

The colSums() function is used to add up the values in each column (each simulation), removing entires with NA. The sum of each column is stored in a vector, which returns the number of occurences of that fraction in each of 1000 simulations. The code to print out the frequency of subsequent common fractions is commented out in order to save space.

```
f_oneHalf <- colSums(binary_12, na.rm = T)
f_oneHalf
```

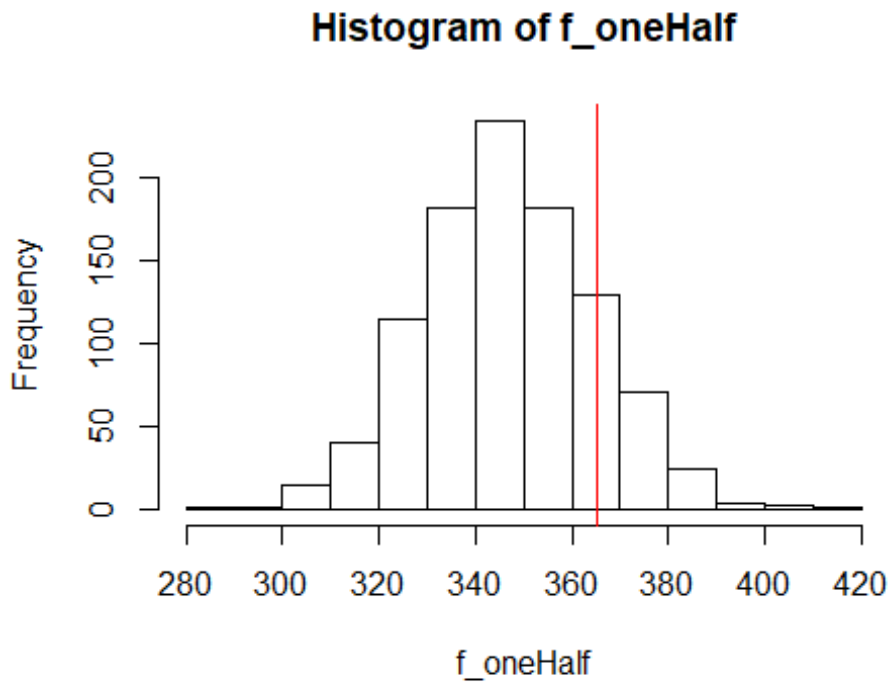
```
##    [1] 341 382 366 360 366 350 328 360 327 376 347 356 331 359 348 367 367
334
##   [19] 330 328 370 367 311 368 347 331 363 319 330 344 366 350 359 365 357
356
##   [37] 343 311 331 327 343 377 413 341 337 374 331 302 322 347 370 381 348
328
##   [55] 318 352 358 392 362 376 318 388 337 367 334 325 368 337 366 356 308
329
##   [73] 357 335 352 356 368 355 324 330 338 367 334 330 357 350 367 375 339
318
##   [91] 328 325 339 369 375 311 345 352 346 343 321 345 347 340 345 341 354
326
##  [109] 337 345 338 367 364 347 349 348 319 369 359 337 323 360 342 344 358
347
##  [127] 335 327 347 358 361 349 338 325 350 329 349 386 348 359 348 342 344
378
##  [145] 337 342 330 327 356 364 357 331 349 324 335 337 347 332 354 322 351
358
##  [163] 339 361 342 358 330 366 375 325 343 334 344 336 351 360 351 327 327
348
##  [181] 367 338 313 360 375 372 345 370 347 345 341 386 343 338 340 375 366
335
##  [199] 352 328 385 329 297 332 363 374 353 362 323 351 358 334 348 355 317
349
##  [217] 331 378 351 355 339 338 387 348 326 362 376 315 357 345 362 349 340
353
##  [235] 335 396 369 341 336 327 321 340 371 348 321 339 320 348 315 359 390
353
##  [253] 319 363 340 342 334 314 350 330 334 349 334 369 342 332 361 336 331
360
##  [271] 325 340 324 348 328 343 328 370 327 307 339 351 353 337 353 320 340
343
##  [289] 342 368 336 341 347 354 353 363 323 355 339 372 331 331 353 350 342
326
##  [307] 335 346 331 389 362 344 344 341 374 359 323 357 343 370 343 379 324
349
##  [325] 370 339 345 313 340 336 350 340 358 336 329 342 309 377 322 353 343
330
##  [343] 343 357 339 317 341 344 336 363 307 316 334 334 359 318 356 364 366
366
##  [361] 335 332 340 324 342 373 386 335 353 314 347 351 337 364 335 351 338
335
##  [379] 329 342 328 336 378 370 358 373 340 329 343 352 322 346 333 374 332
354
##  [397] 341 385 349 321 365 362 347 346 372 370 367 330 342 346 343 325 370
327
##  [415] 351 344 370 368 393 309 305 325 338 319 329 334 327 356 333 339 326
314
##  [433] 342 350 343 340 363 372 354 376 339 354 336 360 350 350 341 348 338
360
```

```
##  [451] 353 354 356 346 353 347 336 373 331 348 343 362 355 347 313 366 355
332
##  [469] 317 346 336 384 340 364 334 362 330 352 377 349 343 325 342 346 358
324
##  [487] 336 348 323 362 359 350 362 361 349 327 333 349 342 329 362 338 354
379
##  [505] 364 373 366 371 373 355 358 349 348 361 357 332 383 341 367 334 389
354
##  [523] 370 361 356 350 368 373 339 360 357 336 351 380 318 355 379 356 365
369
##  [541] 323 350 342 315 335 347 304 343 364 344 355 370 347 357 337 364 341
323
##  [559] 338 357 342 351 362 367 328 353 389 349 341 331 324 357 334 338 357
358
##  [577] 343 343 354 357 371 317 356 352 322 374 336 357 358 327 344 364 345
363
##  [595] 345 342 330 341 353 345 335 373 315 328 341 384 354 340 342 355 372
348
##  [613] 306 357 345 386 370 333 351 336 339 368 345 355 323 366 308 321 339
335
##  [631] 367 350 357 341 355 351 345 286 349 359 339 365 368 364 346 328 349
309
##  [649] 336 330 355 340 345 335 340 318 341 331 369 324 359 339 366 364 339
353
##  [667] 356 337 354 341 339 333 328 360 340 345 344 357 369 335 369 328 350
360
##  [685] 365 335 375 348 363 341 383 371 371 337 374 360 336 332 326 354 338
325
##  [703] 337 342 361 349 309 350 322 368 352 335 342 380 374 349 344 338 332
364
##  [721] 358 370 334 328 358 365 347 349 331 341 354 349 321 371 385 342 361
382
##  [739] 357 363 336 350 354 350 330 344 358 345 374 348 357 321 373 359 352
359
##  [757] 356 334 333 377 347 349 373 351 318 339 332 346 336 326 341 355 372
349
##  [775] 372 338 340 345 350 327 311 349 367 331 357 320 361 368 345 312 339
340
##  [793] 351 377 357 331 376 341 356 357 326 342 371 335 383 352 375 335 358
358
##  [811] 363 337 337 356 335 347 314 339 330 336 354 361 331 352 342 384 343
346
##  [829] 337 388 347 353 371 369 362 400 322 354 332 350 346 347 339 375 346
360
##  [847] 351 325 348 328 349 359 323 356 348 351 337 358 342 344 361 346 338
341
##  [865] 344 360 366 327 372 348 351 324 360 334 341 338 358 326 365 314 348
370
##  [883] 350 350 357 349 309 321 328 319 359 352 368 324 343 341 326 321 380
372
```

```
##  [901] 338 347 356 345 336 380 321 342 320 342 340 380 349 383 379 369 333
337
##  [919] 344 329 342 359 324 374 375 350 330 333 341 347 360 343 356 362 330
338
##  [937] 340 356 366 328 331 355 356 329 350 368 337 364 339 362 345 348 321
363
##  [955] 372 334 344 336 367 349 358 359 375 348 368 408 354 337 363 347 365
353
##  [973] 362 353 341 341 343 340 362 359 357 361 335 354 319 306 371 314 349
352
##  [991] 376 361 405 371 355 356 322 346 349 344
```

A histogram is plotted to reflect how often 1/2 is observed within the simulations. The vertical line represents the observed frequency of this fraction in the 2011 data.

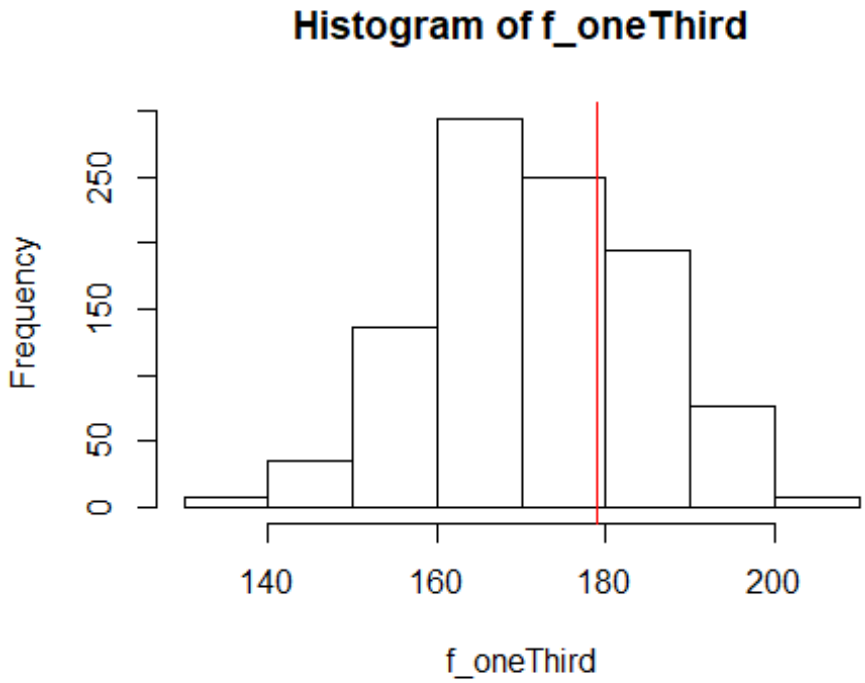
```
hist(f_oneHalf)
r11_one2 <- sum(r11$russiaVS==0.50, na.rm=T)
r11_one2

## [1] 365

abline(v=r11_one2, col="red")
```



COMPARING THE FREQUENCY OF 1/3

The same method for obtaining annd plotting the frequency of 1/2 is applied to finding how often 1/3 occurs in the simulation.

```
binary_13 <- ifelse ((r11_Monte==(1/3)), 1, 0)

f_oneThird <- colSums(binary_13, na.rm = T)
#f_oneThird
hist(f_oneThird)

r11_one3 <- sum(r11$russiaVS==(1/3), na.rm=T)
r11_one3

## [1] 179

abline(v=r11_one3, col="red")
```



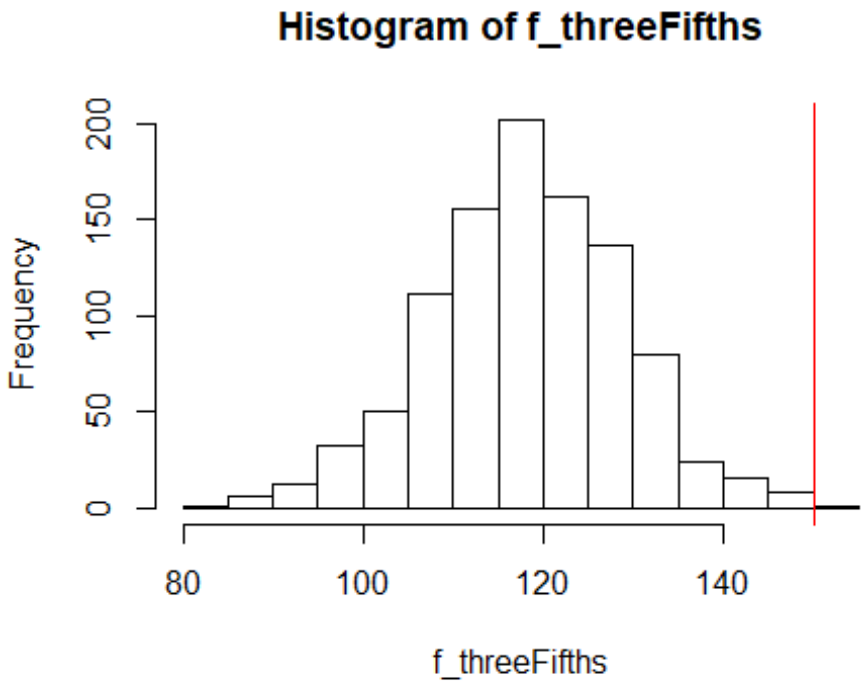
COMPARING THE FREQUENCY OF 3/5

The same method for obtaining and plotting the frequency of 1/2 is applied to finding how often 3/5 occurs in the simulation.

```
binary_35 <- ifelse ((r11_Monte==(3/5)), 1, 0)

f_threeFifths <- colSums(binary_35, na.rm = T)
#f_threeFifths
hist(f_threeFifths)

r11_three5 <- sum(r11$russiaVS==(3/5), na.rm=T)
r11_three5
```

```
## [1] 150
```

```
abline(v=r11_three5, col="red")
```



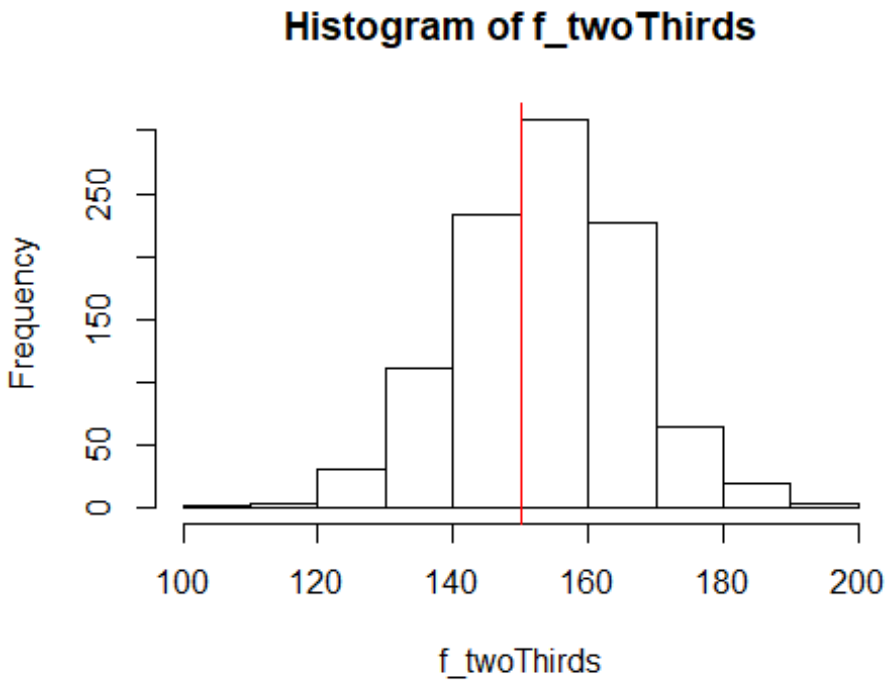
COMPARING THE FREQUENCY OF 2/3

The same method for obtaining and plotting the frequency of 1/2 is applied to finding how often 2/3 occurs in the simulation.

```
binary_23 <- ifelse ((r11_Monte==(2/3)), 1, 0)

f_twoThirds <- colSums(binary_23, na.rm = T)
#f_twoThirds
hist(f_twoThirds)

r11_two3 <- sum(r11$russiaVS==(2/3), na.rm=T)
r11_two3
```

```
## [1] 150
```

```
abline(v=r11_three5, col="red")
```



The vertical lines showing the observed frequency of these common fractions in the 2011 Russian election compared to the histograms do no strongly suggest there was election fraud. The histograms of fraction occurences through the Monte Carlo simulation indicates what the frequency of the fraction is expected to be when there is no fraud. Though the vertical lines are not extremely close to the histogram peaks for all fractions, the line is also not too far off, except in the case for 3/5. For the fractions 1/3 and 2/3, the observed frequencies are actually rather close to the expected frequencies. Cannot confidently conclude there was election fraud.